SD1304
Motorized Vehicles

Advisor: Subbaraya Yuvarajan
Co Advisor: Jeff Erickson
Client: Dr. Selekwa

ECE 403 End Year Presentation

Ross Eickhoff, Nick Aasand, Daniel Zins,
Caleb Walters, Beau Gagnon, Matt Cota

Introduction

Mechanical Engineering design groups need complex electronic design for their projects:

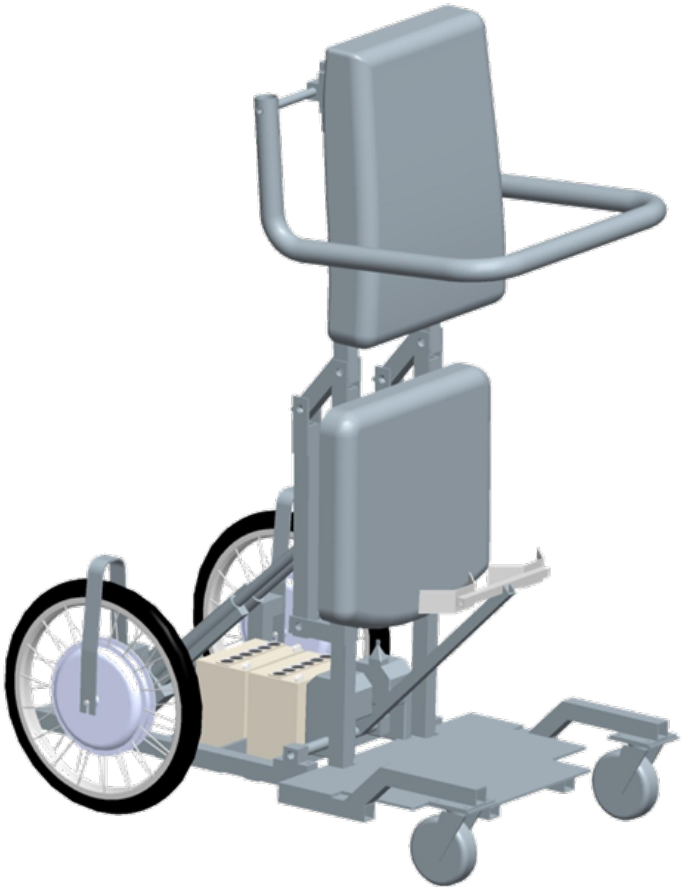
- The autonomous snowplow needs to operate either remotely or by wireless control. The snowplow is designed to compete in the ION Autonomous Snowplow Competition
- The standing wheelchair needs the ability to stand the occupant upright and drive the occupant around using the joystick

Requirements (Wheelchair)

- The standing wheelchair will have a control logic circuit that inputs from a joystick to control the transformation
- The two 24v BLDC motors will be driven by the Roboteq controller w/ joystick input
- The two linear actuators need to be driven to stand up occupant
- The four rotational actuators need to be driven to secure the occupant in

Wheelchair Sitting Position

Wheelchair Standing Position



Requirements (Snow Plow)

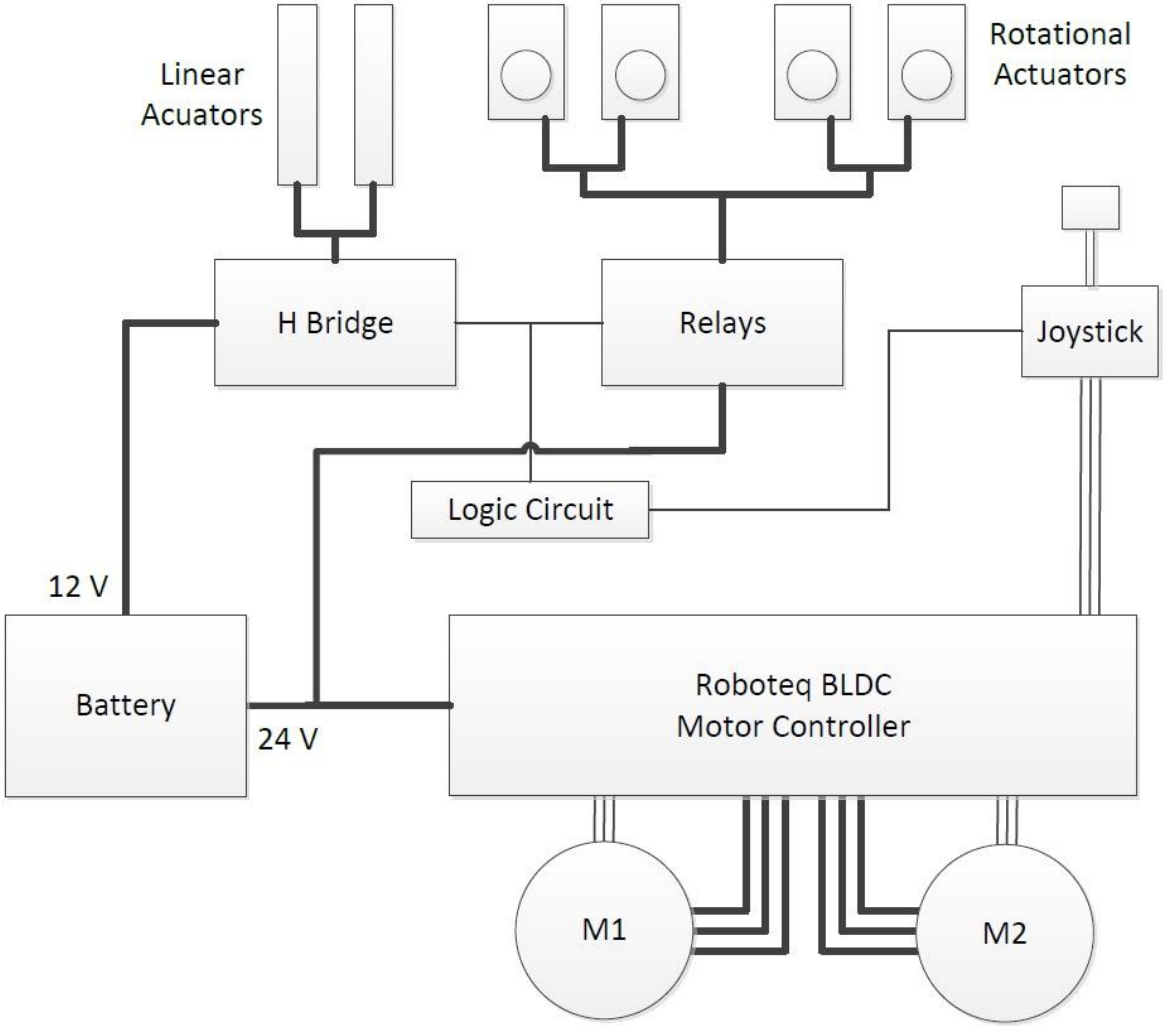
- Use two brushed DC motors, independently for steering
- Drive motors with PWM from Power MOSFET H-bridge
- Operates between 300-1350 RPMs
- Raise and lower plow with winch
- Change pitch of plow with linear actuator
- Steer itself in desired path while avoiding obstacles using GPS and LASER
- Remotely controlled using RF transmitter and receiver
- Run off of 24V battery system

Competition Requirements

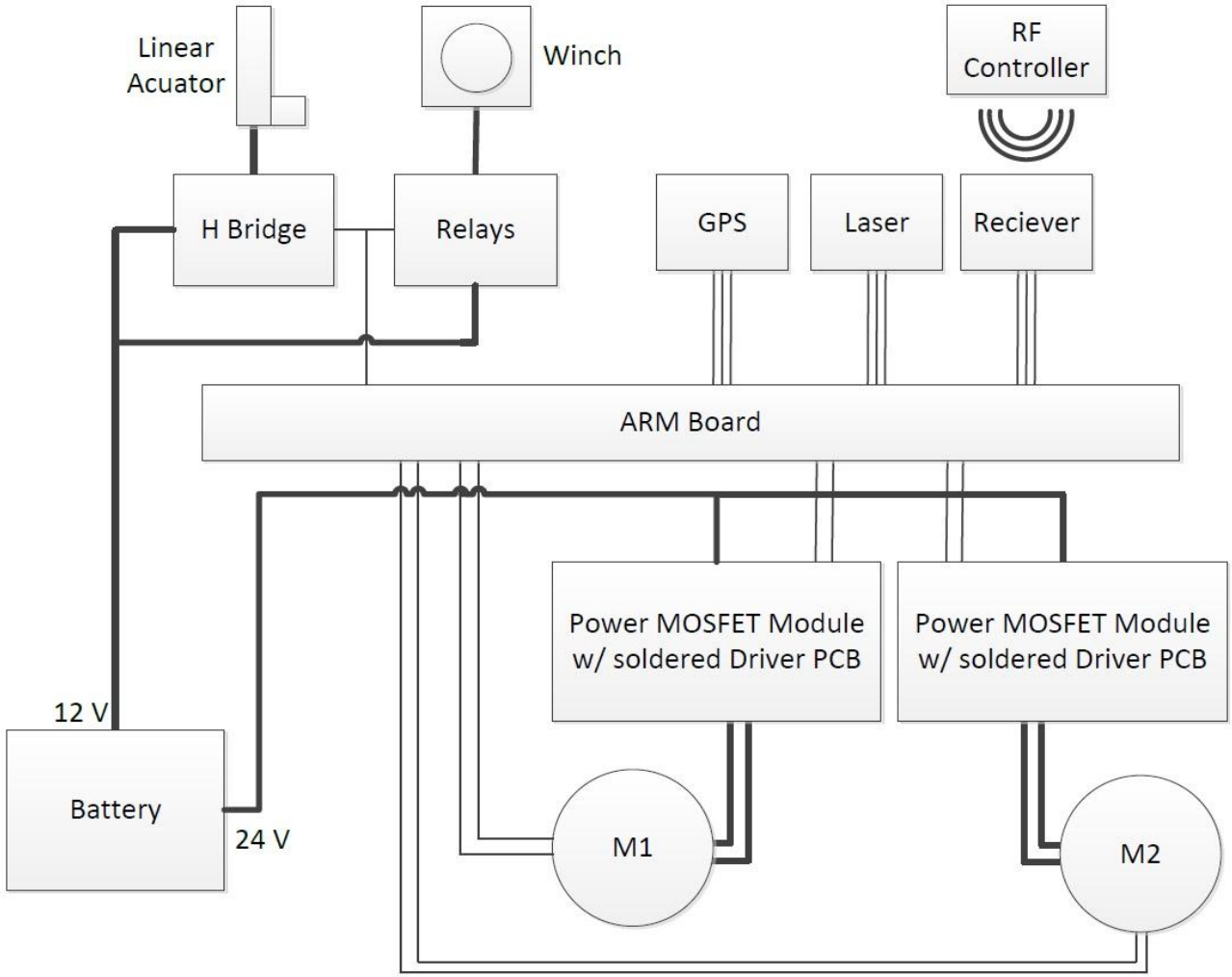
- Two modes: Autonomous and Remote Control
- Emergency stop button on both snowplow and remote that is independent of any other controls
- Must stop within 3 m if emergency stop is pressed
- Remote control must have a minimum range of 50 m for safety
- Max speed of 2 m/s

Snow Plow Model

Standing Wheelchair System

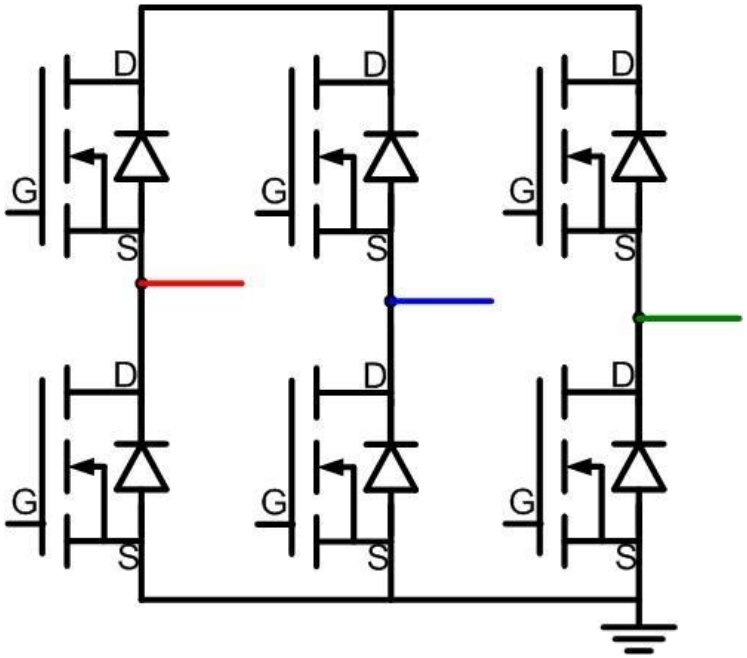


Autonomous Snow Plow System



Brushless DC Motor Drive

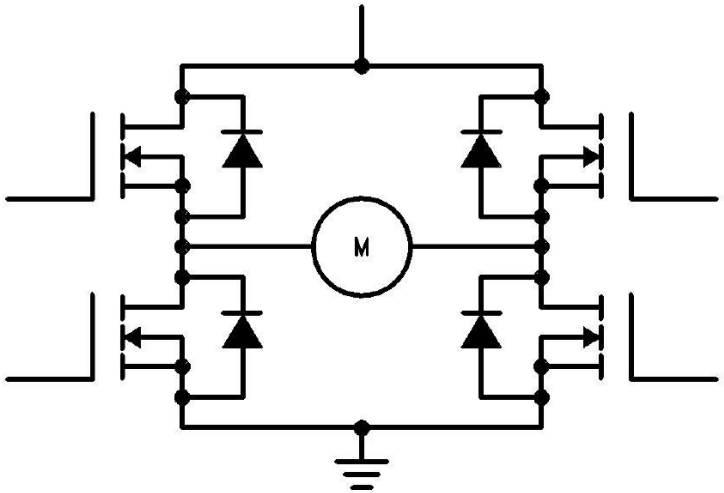
One motor drive



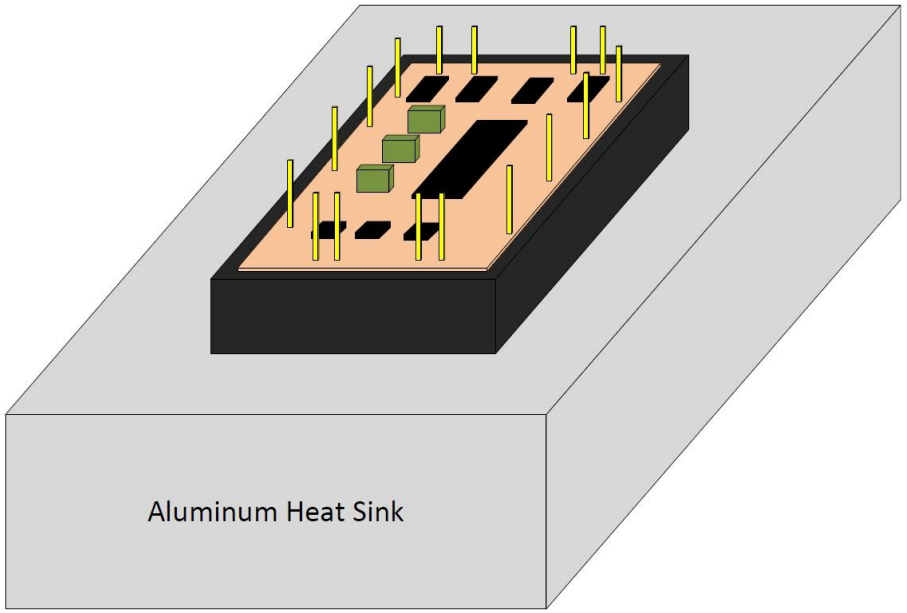
Roboteq VBL 2350

Brushed DC Motor Drive

H-Bridge Configuration



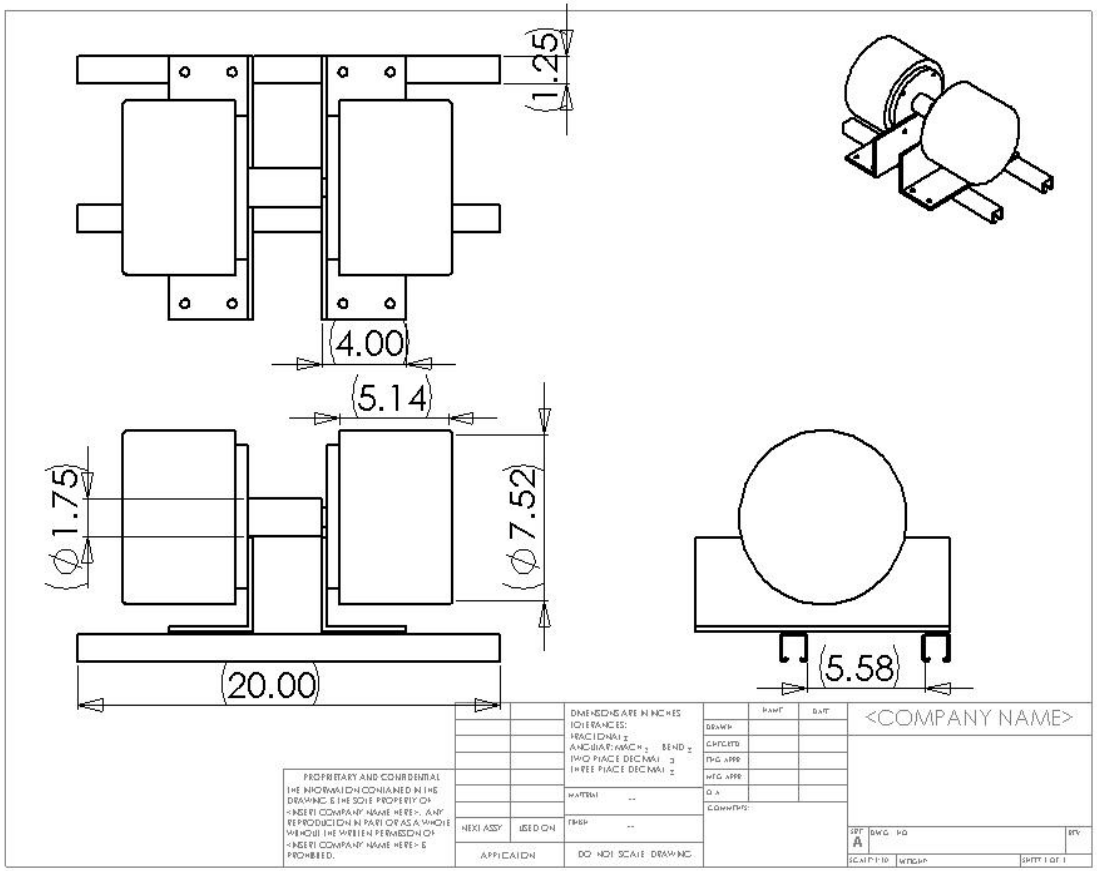
Phase leg model



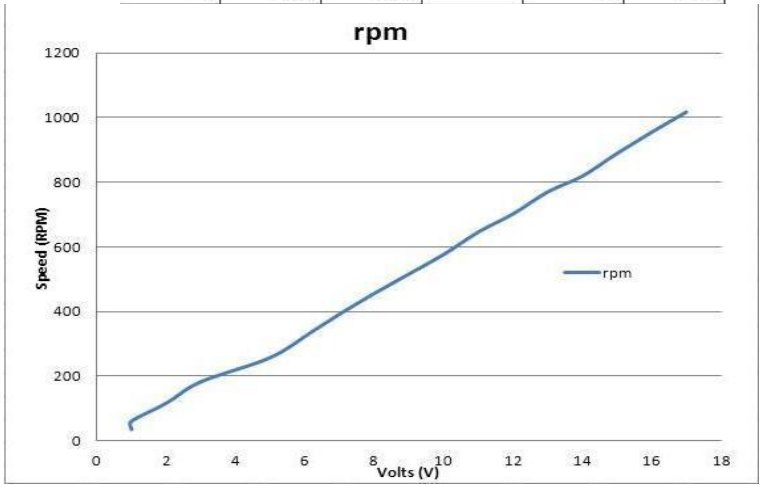
Motor Drive cont.

- The brushed DC motor Drive is capable of driving high current loads in excess of 100 amps continuous
- The high current terminals on the bridge will be connected by point-to-point soldering
- Heatsinking will be required to ensure the longevity of the motor drive
- A full bridge MOSFET gate driver will provide the interface from the controller to the mcu

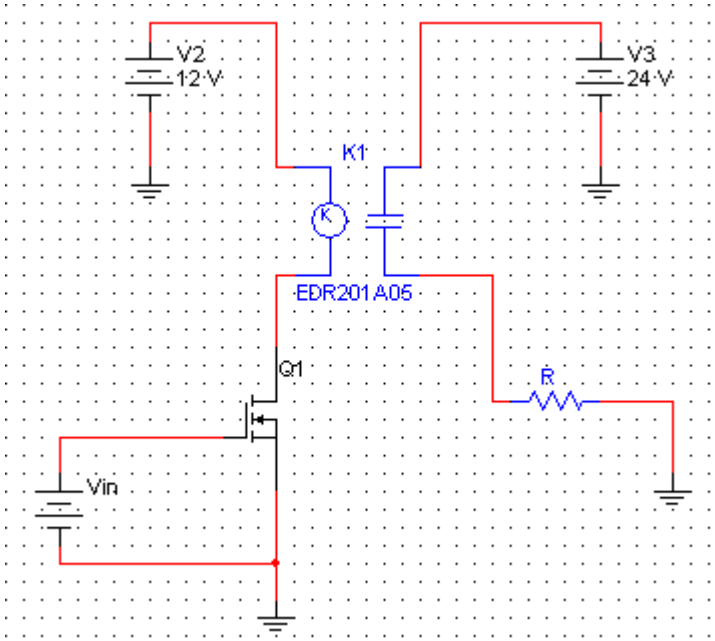
Motor Testing

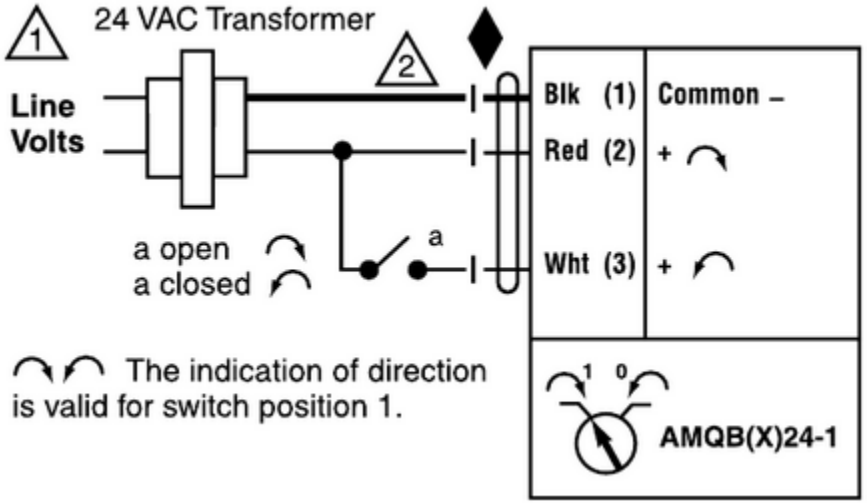


Motor				Generator	
Volts	Current	rpm		volts	Current
1	9	34.8			
1	10	61			
2	12	116.5			
3	14	182.8			
5	16	257.5			
6.2	18	337.3			
7	19.2	392			
8	20.7	456		5.7	8.6
9	22.2	516		6.6	9.7
10	23.7	576.4		7.4	10.9
11	25.3	646		8.2	12.1
12	26.8	702		9	13.2
13	28.3	770		9.8	14.5
14	29.6	819			
15	31.3	889		11.3	16.6
16	33	955		12.2	17.8
17	34.4	1017		13	18.9



Relay Board


V2
12 V
V3
24 V
K1
EDR201A05
Q1
R
Vin

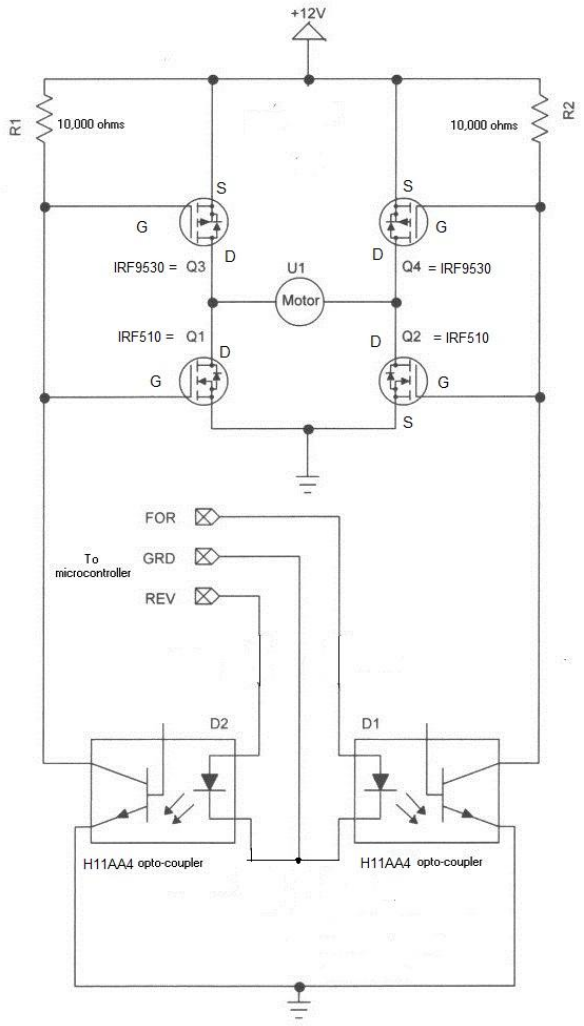

1
24 VAC Transformer
2
Line Volts
Blk (1)
Common –
Red (2)
+
Wht (3)
+
a
a open
a closed
The indication of direction is valid for switch position 1.
1
0
AMQB(X)24-1

H-Bridge

Utilizes 2 N and 2 P channel Logic Level Mosfets

Opto Isolators to prevent damage to ARM

May utilize Allegro A4940 Controller if using all N channel



NXP LPC1769 Xpresso (ARM)

PWM H-bridge Program in C Code

```
Loop Indefinate
        Set count = 100;
        Set P1,2,3,4 =0
        If button 1 pressed
                count =count+1
        If button 2 =1
                count=count-1
        If count < 100
                If count>0
                        turn P1,2 on/ P3,4 off
                        Wait for 100-count
                        Turn P1 off
                         Wait for count
```

NXP LPC1769 Xpresso (ARM)

If count >100

If count <200

    turn P3,4 on/ P1,2 off

    Wait for count-100

    Turn P4 off

    Wait for 200-count

If count =100

    Turn P1,2,3,4 off

If button 3 =1

    Count=100

    Turn P1,2,3,4 off

PWM C Code for H-Bridge

```
#include "LPC17xx.h"
int get_pause(int time)
{
	int I;
	if((LPC_GPIO0->FIOPIN &= (1<<2)) == 0) // P0.2 pushed
	{
		if((time<=200))
		{
		time=time+1;
		for (I=0;I<(100000);I++);
		}
	}
	if((LPC_GPIO0->FIOPIN &= (1<<3)) == 0) //P0.3 pushed
	{
		if((time>=0))
		{
			time=time-1;
			for (I=0;I<(100000);I++);
		}
	}
return time;
}
```

PWM C Code for H-Bridge

```
int PWM_forward (int time)
{
        int I,width;
        while(time>100)
        {
                time=get_pause(time);
                width=time-100;

                LPC_GPIO1 -> FIODIR |= (12<<19);
                LPC_GPIO1 -> FIOPIN = (12<<19);

                LPC_GPIO1 -> FIODIR |= (3<<19);
                LPC_GPIO1 -> FIOPIN &= ~(3<<19);            //output on
                for (I=0;I<(width);I++);
                LPC_GPIO1 -> FIODIR |= (1<<19);
                LPC_GPIO1 ->FIOPIN ^= (1<<19);                      //output off
                for (I=0;I<((100-width));I++);                      //P0.2 pause/start
        }
return (time);
}
```

PWM C Code for H-Bridge

```
int PWM_reverse(int time)
{
int I,width;
        while(time<100)
        {
                time=get_pause(time);
                width=100-time;
                if(width<200)
                {
                LPC_GPIO1 -> FIODIR |= (3<<19);
                LPC_GPIO1 -> FIOPIN = (3<<19);
                LPC_GPIO1 -> FIODIR |= (12<<19);
                LPC_GPIO1 -> FIOPIN &= ~(12<<19);          //output on
                for (I=0;I<width;I++);
                LPC_GPIO1 -> FIODIR |= (8<<19);
                LPC_GPIO1 ->FIOPIN ^= (8<<19);                  //output off
                for (I=0;I<(100-width);I++);                //P0.2 pause/start
                }
        }
return (time);
}
```

PWM C Code for H-Bridge

```
int stop(int time)
{
        time=100;
        time=get_pause(time);
        LPC_GPIO1 -> FIODIR |= (0000<<19);
        LPC_GPIO1 -> FIOPIN ^= (0000<<19);
return (time);
}


int reset_PWM(int time)
{
time =100;
LPC_GPIO0 -> FIODIR |= (0000<<19);
LPC_GPIO0 -> FIOPIN ^= (0000<<19);
return time;
}
```

PWM C Code for H-Bridge

```
int main ()
{
int time=0;
time=stop(time);
while(1)
        {
                if (time==100)
                {
                        time=get_pause(time);
                        time=stop(time);
                }
                else if (time<100)
                {
                        time=get_pause(time);
                        time=PWM_reverse(time);
                }
                else if (time>100)
                {
                        time=get_pause(time);
                        time=PWM_forward(time);
                }
```

PWM C Code for H-Bridge

```
                else
                {
                        time=stop(time);
                }
                if((LPC_GPIO0->FIOPIN &= (1<<4)) == 0)
                {
                        time=reset_PWM(time);
                }
        }
return(1);
}
```

Copernicus II GPS Module

- GPS will be used for destinations and current location of the autonomous snowplow
- Updates at a rate of 1 Hz
- 90% accurate within 4 meters
- We will be using two modules for increased accuracy
- Each destination will be recorded onto microcontroller

Laser

- Will be used to detect obstacles in the way of the snowplow (for competition and safety)
- Sends angle and distance of obstacle to microcontroller
- Is not primary method for driving the snowplow but will contribute

Project Status

Wheelchair

- Get Motor Controller working
- Develop Logic Control
- Get H-bridge and relay to layout
- Test and Implement into system

Snow Plow

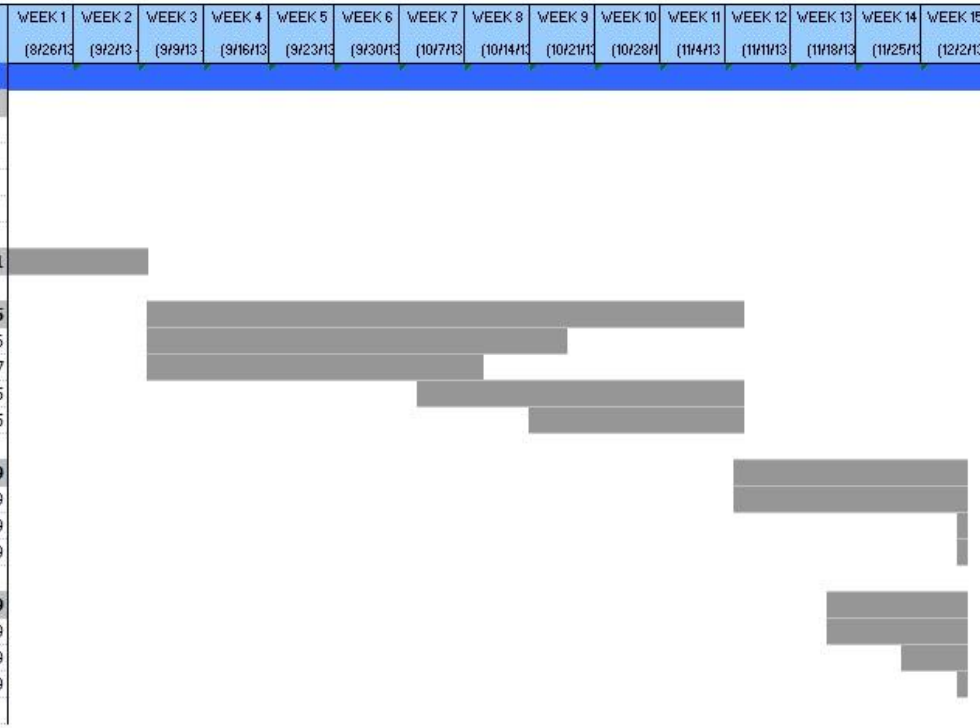
- Touch up motor drive code
- Develop and build MOSFET module and driver board
- Develop sensory code to receive inputs from system
- Research and develop GPS and LASER systems
- Develop and build RF receiver

Gantt Chart

SENIOR DESIGN III

- **Project Name** : Motorized Vehicles
- **Project Description** : Developing Dev boards for ME Design groups
- **Project Length** : 1 year
- **Start Date** : 26-Aug-13 — 9-Dec-13 / 15
- **Working Days** : Monday - Sunday
- **Today's Marker** : Yes
- **Holiday's Marker** : Yes

Level	Task	PIC	Start Date	Finish Date	WD	DC	DR
1	**Summer**						
1.1	BLDC motor controller working and interfaced						
1.2	Get power MOSFET driver built and Drive module						
1.3	Get power MOSFET driver PCB built						
1.4	Get relay and h-bridge PCB built						
2	**Revise Budget, Timeline, Scope**	**All**	**26-Aug-13**	**9-Sep-13**	**15**	**##**	**131**
3	**Develop Input systems and circuits**	**All**	**10-Sep-13**	**12-Nov-13**	**64**	**##**	**195**
3.1	LASER and GPS	Nick, Matt	10-Sep-13	24-Oct-13	45	-131	176
3.2	RF Controller	Beau, Ross	10-Sep-13	15-Oct-13	36	-131	167
3.3	Code to receive and act upon input	Dan, Caleb	9-Oct-13	12-Nov-13	35	-160	195
3.4	Finish driver circuit	Ross, Nick	21-Oct-13	12-Nov-13	23	-172	195
4	**Implementation and Physical Build**	**All**	**12-Nov-13**	**6-Dec-13**	**25**	**##**	**219**
4.1	Layout for extra input circuitry	Matt, Beau, Dan	12-Nov-13	6-Dec-13	25	-194	219
4.2	Implement Processor with other PCB	All	6-Dec-13	6-Dec-13	1	-218	219
4.3	Troubleshoot whole system.	All	6-Dec-13	6-Dec-13	1	-218	219
5	**Implementation and Presentation**	**All**	**22-Nov-13**	**6-Dec-13**	**15**	**##**	**219**
5.1	Install boards on snow plow and troubleshoot	Matt, Beau, Dan	22-Nov-13	6-Dec-13	15	-204	219
5.2	Put presentation together and finish documents	All	30-Nov-13	6-Dec-13	7	-212	219
5.3	Design presentation	All	6-Dec-13	6-Dec-13	1	-218	219



Budget

Shipping	Shipping	1	$10.95	$10.95
2327	Concentric LACT12P-12V-5 Linear Actuator with Feedback: 12"	1	$109.95	$109.95
Shipping	Shipping	1	$7.99	$7.99
924-EA-XPR-003	Development Boards & Kits - ARM LPCXPRESSO LPC1769	2	$29.87	$59.74
Shipping	Shipping Order#84101470	1	$0.00	$0.00
719302	Rover App- Controlled Spy Tank	1	$79.99	$79.99
HIP4086APZ-ND	IC DRIVER FET 3PHASE N-CH 24DIP	2	$6.75	$13.50
Shipping	Shipping USPS	1	$6.38	$6.38
4SJ300-B-M2-SB	3 Axis Joystick	1	$46.00	$46.00
PB1132-ND	RELAY GEN PURPOSE SPST 10A 12V	4	$1.42	$5.68
AMQB24-1	SPST Power Relay Actuator	1	$223.00	$223.00
Total				**$563.18**

Summary

- It has been tough working for the ME groups, from delays to requirement changes.
- This has lead to a more systematic design and optimization on a higher level.
- We have had to change some of our ideas and implement new ones to save time and complexity
- It was a slow start, but we are now moving forward into the summer and next fall